

NETWORK PROTOCOLS

After reading this chapter and completing the exercises, you will be able to:

- Identify the characteristics of TCP/IP, IPX/SPX, NetBIOS, and AppleTalk
- Understand the position of network protocols in the OSI Model
- Identify the core protocols of each protocol suite and its functions
- Understand each protocol's addressing scheme
- Install protocols on Windows 98 and Windows 2000 clients



ON THE JOB

I work for a company that remotely monitors network environments for other companies. We gather statistics on everything from when a server goes down to when the humidity in a data center drops too low. We use this data to alert our clients to situations that may affect their networks and their bottom lines.

One evening our network monitoring system received a steady stream of alerts from one of our client's networks, indicating that something was wrong. Upon further investigation, I traced the alerts to a server that, for some reason, was not responding to requests. This server provided the main Web site for retail customers, so any downtime meant lost revenue for our client. I tried pinging the server by name, and indeed, it did not respond. Then I tried pinging its IP address, and I did receive a response—but not from the server. Instead, the response was from a workstation also located at the client site.

I called the client's network administrator and told him what I had discovered. He located the workstation that was responding to the IP address. Although this IP address was supposed to belong to the server, a new employee had inadvertently assigned his own workstation this IP address. Some time after the employee did so, the server had been rebooted and could not use its given IP address. Thus, it was not responding to any requests. The client's network administrator immediately disconnected the offending workstation from the network and rebooted the Web server so that it could once again use its correct IP address.

Roger DuRocher
Full Control Systems

As you learned in Chapter 1, a **protocol** is a rule that governs how networks communicate. Protocols define the standards for communication between network devices. Without protocols, devices could not interpret the signals sent by other devices, and data would go nowhere. Unfortunately, you cannot turn on a file server, add some clients, and expect the protocols to work their magic. Instead, you must first understand which protocol suits your network environment. Then you must install and configure protocols on file servers and clients and test your configuration.

In this chapter, you will learn about the most commonly used networking protocols, their components, and their functions. This chapter is not an exhaustive study of protocols, but rather a practical guide to applying them. At the end of the chapter, you will have the opportunity to read about some realistic networking scenarios pertaining to protocols and devise your own solutions. As protocols form the foundation of network communications, you must fully understand them to manage a network effectively.

INTRODUCTION TO PROTOCOLS

In Chapter 2, you learned about the tasks associated with each layer of the OSI Model. These tasks are actually carried out by network protocols. In the networking industry, the term “protocol” is often used to refer to a group, or suite, of individual protocols that work together. The protocols within a suite are assigned different tasks, such as data translation, data handling, error checking, and addressing; they correspond to different layers of the OSI Model. In the sections that follow, you will learn about the four major networking protocol suites—TCP/IP, IPX/SPX, NetBIOS, and AppleTalk—and see how their components correspond to the layers of the OSI Model. You must understand these protocols to qualify for Net+ certification. Pay particular attention to the TCP/IP discussions, because the Net+ certification exam emphasizes TCP/IP knowledge.

The protocol (or protocol suite) you use will depend on many factors, including the existing network operating environment, your organization’s technical expertise, and your network’s security and speed requirements. Protocols vary according to their speed, transmission efficiency, utilization of resources, ease of setup, compatibility, and ability to travel between one LAN segment and another. Protocols that can span more than one LAN segment are **routable**, because they carry Network layer and addressing information that can be interpreted by a router. Not all protocols are routable, however.


In addition to the size of the network, you will need to consider its interconnection requirements, data security needs, and the technical expertise of personnel who manage the network. Many networks use more than one kind of protocol because they have a mixed hardware or software infrastructure, so it is not only important to know about each protocol, but also to understand how they work together. A network that uses more than one protocol is called a **multiprotocol network**. Multiprotocol networks are common in businesses whose LANs are well established and have evolved from legacy systems to newer, more efficient networks.

As you read in this chapter about the most commonly used protocols, keep in mind that you may occasionally encounter additional protocols (such as SNA or DLC) on a network. The more flexible and robust protocols described in this chapter are gradually replacing these older protocols. TCP/IP is by far the most commonly used of the major protocols, followed by IPX/SPX, then NetBIOS and AppleTalk. In the next section, you’ll begin by learning about the most popular of the four—TCP/IP.

TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL (TCP/IP)

TCP/IP is not simply one protocol, but rather a suite of small, specialized protocols—including TCP, IP, UDP, ARP, ICMP, and others—called **subprotocols**. Most network administrators refer to the entire group as “TCP/IP,” or sometimes simply “IP.” TCP/IP’s roots lie with the U.S. Department of Defense, which developed the precursor to TCP/IP for its Advanced Research Projects Agency network (ARPAnet) in the late 1960s. Thanks to its low cost and its ability to communicate between a multitude of dissimilar platforms, TCP/IP has grown extremely popular. It is a de facto standard on the Internet and is fast becoming the protocol of choice on LANs. The latest network operating systems (such as NetWare 5.x and Windows 2000) use TCP/IP as their default protocol.

One of the greatest advantages to using TCP/IP relates to its status as a routable protocol, which means that it carries network addressing information that can be interpreted by routers. TCP/IP is also a flexible protocol, running on any combination of network operating systems or network media. Because of its flexibility, however, TCP/IP may require significant configuration.



TCP/IP is a broad topic with numerous theoretical, historical, and practical aspects. Because it is such an important protocol, it is covered in even more detail in Chapter 11. If you want to become an expert on TCP/IP, you should invest in a book or study guide solely devoted to this suite of protocols.

TCP/IP Compared to the OSI Model

The TCP/IP suite of protocols can be divided into four layers that roughly correspond to the seven layers of the OSI Model, as depicted in Figure 3-1 and described in the following list.

OSI Model		TCP/IP Model	
Application		Application	
Presentation			
Session			
Transport		Transport	
Network		Internet	
Data Link		Network Interface	
Physical			

Figure 3-1 TCP/IP compared to the OSI Model

- *Application layer*—Roughly equivalent to the Application, Presentation, and Session layers of the OSI Model. Applications gain access to the network through this layer, via protocols such as the File Transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP), Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), and Dynamic Host Configuration Protocol (DHCP).
- *Transport layer*—Roughly corresponds to the Transport layer of the OSI Model. This layer holds the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), which provide flow control, error checking, and sequencing. All service requests use one of these protocols.
- *Internet layer*—Equivalent to the Network layer of the OSI Model. This layer holds the Internet Protocol (IP), Internet Control Message Protocol (ICMP), Internet Group Message Protocol (IGMP), and Address Resolution Protocol (ARP). These protocols handle message routing and host address resolution.
- *Network Interface layer*—Roughly equivalent to the Data Link and Physical layers of the OSI Model. This layer handles the formatting of data and transmission to the network wire.

The TCP/IP Core Protocols

Certain subprotocols of the TCP/IP suite, called **TCP/IP core protocols**, operate in the Transport or Network layers of the OSI Model and provide basic services to the protocols in other layers of the four-layer model. As you might guess, TCP and IP are the most significant core protocols in the TCP/IP suite. These, plus some other core protocols are discussed below.

Internet Protocol (IP)

The **Internet Protocol (IP)** belongs to the Internet layer of the TCP/IP Model and provides information about how and where data should be delivered. IP is the subprotocol that enables TCP/IP to **internetwork**—that is, to traverse more than one LAN segment and more than one type of network through a router. In an internetwork, the individual networks that are joined together are called subnetworks, or **subnets**. Using subnets is an important part of TCP/IP networking.



The following sections describe the IP subprotocol as it is used in IP version 4 (IPv4), the original version that was used for nearly 20 years and is still used by most networks today. A newer version of the IP subprotocol, called IP version 6 (IPv6), will soon replace IPv4.

The IP portion of a data frame is called an **IP datagram**. The IP datagram acts as an envelope for data and contains information necessary for routers to transfer data between subnets. The length of the IP datagram including its header and data cannot exceed 65,535 bytes. The components of an IPv4 IP datagram header are described in the following list and depicted in Figure 3-2.

- *Version*—Identifies the version number of the protocol. The receiving workstation looks at this field first to determine whether it can read the incoming data. If it cannot, it will reject the packet. Rejection rarely occurs, however, because most TCP/IP networks use IP version 4 (IPv4). A more sophisticated IP version, called IP version 6 (IPv6), has been developed and will be implemented in coming years.
- *Internet header length (IHL)*—Identifies the number of 4-byte (or 32-bit) blocks in the IP header. The most common header length comprises five groupings, as the minimum length of an IP header is 20 4-byte blocks. This field is important because it indicates to the receiving node where data will begin (immediately after the header ends).
- *Type of service (ToS)*—Tells IP how to process the incoming datagram by indicating the data's speed, priority, or reliability.
- *Total length*—Identifies the total length of the IP datagram, including the header and data, in bytes.
- *Identification*—Identifies the message to which a datagram belongs and enables the receiving node to reassemble fragmented, or segmented, messages. This field and the following two fields, flags and fragment offset, assist in segmentation and reassembly of packets.
- *Flags: don't fragment (DF) or more fragments (MF)*—Indicates whether a message is fragmented and, if it is fragmented, whether the datagram is the last in the fragment.
- *Fragment offset*—Identifies where the datagram fragment belongs in the incoming set of fragments.
- *Time to live (TTL)*—Indicates the maximum time, in seconds, that a datagram can remain on the network before it is discarded. TTL also corresponds to number of router hops that a datagram can go through; each time a datagram passes through a machine, another second is taken off its TTL, regardless of whether the machine took a whole second to process the data.
- *Protocol*—Identifies the type of Transport layer protocol that will receive the datagram (for example, TCP or UDP).
- *Header checksum*—Determines whether the IP header has been corrupted.
- *Source IP address*—Identifies the full IP address of the source node.
- *Destination IP address*—Indicates the full IP address of the destination node.
- *Options*—May contain optional routing and timing information.
- *Padding*—Contains filler information to ensure that the header is a multiple of 32 bits. The size of this field may vary.
- *Data*—Includes the data originally sent by the source node, plus TCP information.

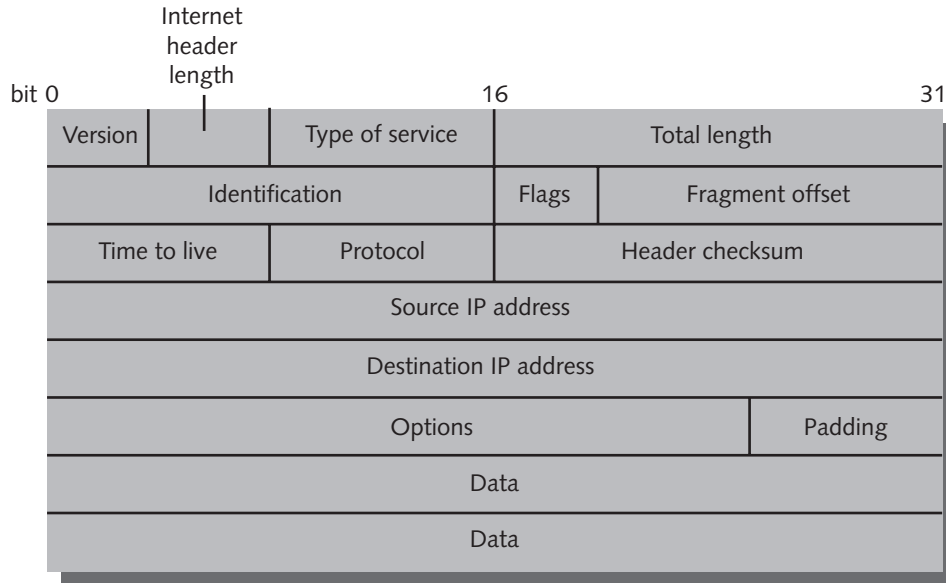


Figure 3-2 Components of an IP datagram

IP is an unreliable, **connectionless** protocol, which means that it does not guarantee delivery of data. Higher-level protocols of the TCP/IP suite, however, can use IP information to ensure that data packets are delivered to the right addresses. Note that the IP datagram does contain one checksum component, the header checksum, which verifies only the integrity of the routing information in the IP header. If the checksum accompanying the message does not have the proper value when the packet is received, then the packet is presumed to be corrupt and is discarded; at that point, a new packet is sent.

Transmission Control Protocol (TCP)

The **Transmission Control Protocol (TCP)** belongs to the Transport layer of the TCP/IP suite and provides reliable data delivery services. TCP is a **connection-oriented** subprotocol, which means that a connection must be established between communicating nodes before this protocol will transmit data. TCP sits on top of the IP subprotocol and compensates for IP's reliability deficiencies by providing checksum, flow control, and sequencing information. If an application relied only on IP to transmit data, IP would send packets indiscriminately, without checking whether the destination node is offline, for example, or whether the data becomes corrupt during transmission. TCP, on the other hand, contains several components that ensure data reliability. The fields of the **TCP segment**, the entity that becomes encapsulated by the IP datagram, are described in the following list. Figure 3-3 depicts a TCP segment and its fields.

- *Source port*—Indicates the port number at the source node. A **port** is the address on a host where an application makes itself available to incoming

data. One example of a port is port 80, which is typically used to accept Web page requests. You will learn more about ports in Chapter 11.

- *Destination port*—Indicates the port number at the destination node.
- *Sequence number*—Identifies the data segment's position in the stream of data segments already sent.
- *Acknowledgment number (ACK)*—Confirms receipt of the data via a return message to the sender.
- *TCP header length*—Indicates the length of the TCP header.
- *Codes*—Includes flags that signal special conditions—for example, if a message is urgent, or if the source node wants to request a connection or terminate a connection.
- *Sliding-window size*—Indicates how many blocks of data the receiving machine can accept.
- *Checksum*—Allows the receiving node to determine whether the TCP segment became corrupted during transmission.
- *Urgent pointer*—Can indicate a location in the data where urgent data resides.
- *Options*—Used to specify special options.
- *Padding*—Contains filler information to ensure that the size of the TCP header is a multiple of 32 bits.
- *Data*—Contains data originally sent by the source node.

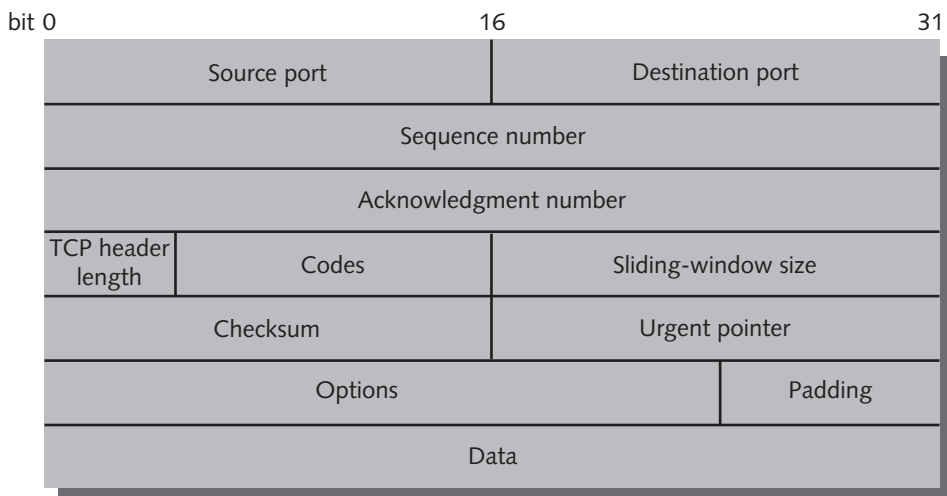


Figure 3-3 A TCP segment

User Datagram Protocol (UDP)

The **User Datagram Protocol (UDP)**, like TCP, sits in the Transport layer, between the Internet layer and the Application layer of the TCP/IP model. Unlike TCP, however, UDP is a connectionless transport service. UDP offers no assurance that packets will be received in the correct sequence. In fact, this protocol does not guarantee that the packets will be received at all. Furthermore, it provides no error checking or sequence numbering. Nevertheless, UDP's lack of sophistication makes it more efficient than TCP and renders it useful in situations where data must be transferred quickly, such as live audio or video transmissions over the Internet. In these cases, TCP—with its acknowledgments, checksums, and flow control mechanisms—would add too much overhead to the transmission and bog it down. In contrast to TCP's 10 fields, the UDP header contains only four fields: source port, destination port, length, and checksum.

Internet Control Message Protocol (ICMP)

Whereas IP ensures that packets reach the correct destination, **Internet Control Message Protocol (ICMP)** notifies the sender when something goes wrong in the transmission process and the packets are not delivered. ICMP sits between IP and TCP in the Internet layer of the TCP/IP model and does not provide error control. Instead, it simply reports which networks are unreachable and which packets have been discarded because the allotted time for their delivery (their TTL) expired. ICMP is used by diagnostic utilities such as PING and TRACERT, which are described in Chapter 11.

Address Resolution Protocol (ARP)

Address Resolution Protocol (ARP) is an Internet layer protocol that obtains the MAC (physical) address of a host, or node, then creates a local database that maps the MAC address to the host's IP (logical) address. ARP works very closely with IP, because IP must have the address of a destination host before it can direct data to it. If one host needs to know the MAC address of another host on the same subnet, the first host sends a broadcast message to the network through ARP that essentially says, "Will the computer with the IP address AA.BB.CC.DD please send me its MAC address?" The host on the local subnet that has the IP address AA.BB.CC.DD then broadcasts a reply that contains the physical address of the destination host. To make ARP more efficient, computers save recognized IP-to-MAC address mappings in a cache, so they don't have to broadcast redundant requests.

The TCP/IP Application Layer Protocols

In addition to the core Transport and Internet layer protocols, TCP/IP encompasses several Application layer protocols. These protocols work over TCP or UDP and IP, translating user requests into a format the network can read. The following list describes the most commonly used Application layer protocols:

- *Telnet*—A terminal emulation protocol used to log on to remote hosts using the TCP/IP protocol suite. Often Telnet is used to connect two dissimilar

systems (such as PCs and UNIX machines). Through Telnet, you can control a remote host over LANs and WANs such as the Internet. For example, network managers can use Telnet to log on to their company's routers from home and modify the router's configuration.

- *File Transfer Protocol (FTP)*—A protocol used to send and receive files via TCP/IP. FTP is a client/server protocol in which the host running the FTP server portion accepts commands from another host running the FTP client portion. It comes with a set of very simple commands that make up its user interface.
- *Simple Mail Transfer Protocol (SMTP)*—The protocol responsible for moving messages from one e-mail server to another over the Internet and other TCP/IP-based networks. SMTP uses a simple request-and-response mechanism to move messages and relies upon more sophisticated protocols, such as the Post Office Protocol (POP), to keep track of storing and forwarding messages.
- *Simple Network Management Protocol (SNMP)*—A communication protocol used to manage devices on a TCP/IP network. To use SNMP, each device on the network runs an agent that collects information about that device. SNMP transports the collected information to a central database. Many network management programs use SNMP.

You will learn about more Application layer TCP/IP protocols in Chapter 11.

Addressing in TCP/IP

As you learned in Chapter 1, each node on a network must have a unique identifying number called an address. You have also learned that networks recognize two kinds of addresses: logical and physical (or MAC) addresses. MAC addresses are assigned to a device's network interface card at the factory by its manufacturer, but logical addresses depend on rules set by the protocol standards. In the TCP/IP protocol suite, IP is the core protocol responsible for logical addressing. For this reason, addresses on TCP/IP networks are sometimes called "**IP addresses**." IP addresses are assigned and used according to very specific parameters.

Each IP address is a unique 32-bit number, divided into four **octets**, or 8-bit bytes, that are separated by periods. An example of a valid IP address is 144.92.43.178. An IP address contains two types of information: network and host. The first octet identifies the network class. Three types of network classes exist: Class A, Class B, and Class C. Table 3-1 summarizes the three commonly used classes of TCP/IP networks.

Table 3-1 Commonly used TCP/IP classes

Network Class	Beginning Octet	Number of Networks	Host Addresses per Network
A	1–126	126	16,777,214
B	128–191	>16,000	65,534
C	192–223	>2,000,000	254



Although 8 bits have 256 possible combinations, only the numbers 1 through 254 can be used to identify networks and hosts. The numbers 0 and 255 are reserved for **broadcasts**, or transmissions to all stations on a network.

All nodes on a Class A network share the first octet of their IP numbers, a number between 1 and 126. Nodes on a Class B network share the first two octets, and their IP addresses begin with a number between 128 and 191. Class C network IP numbers share the first three octets, with their first octet being a number between 192 and 223. For example, nodes with the following IP addresses may belong to the same Class A network: 23.78.110.109, 23.164.32.97, 23.48.112.43, and 23.108.37.22. Nodes with the following IP addresses may belong to the same Class B network: 168.34.88.29, 168.34.55.41, 168.34.73.49, and 168.34.205.113. Nodes with the following addresses may belong to the same Class C network: 204.139.118.7, 204.139.118.54, 204.139.118.14, and 204.139.118.31.

Because only 126 Class A networks are available on the Internet, most Class A networks have already been reserved by large corporations or governments. In addition, some IP addresses are reserved for network functions, like broadcasts, and cannot be assigned to machines or devices. Notice that 127 is not a valid first octet for any IP number. The range of addresses beginning with 127 is reserved for loopback information, with the IP address 127.0.0.1 being called a **loopback address**. When you try to contact this IP number, you are actually communicating with your own machine. This address can prove useful when you must troubleshoot problems with a workstation's TCP/IP communications. If you receive a positive response from the loopback test, you know that the TCP/IP protocols are installed and in use on your workstation.

A company can request a class of network addresses from the **Internet Corporation for Assigned Names and Numbers (ICANN)**, the non-profit corporation currently designated by the U.S. government to maintain and assign IP addresses or an agent (such as an Internet service provider (ISP) that will either request network addresses from ICANN on the company's behalf or lease some of its already-reserved IP addresses to the company.

For example, suppose that you decide to go into business for yourself, selling high-quality, homemade jams and jellies on the Internet. You name your company "Jan's Jams," and you hire five staff members who will work on your office network in your shop. You would like each staff person to have his or her own address to use when communicating with the Internet. You also think that your products will be so successful that you might add five or more new staff members in the coming year. To obtain addresses for all existing staff and allow for growth, you can register with ICANN for a group of 16 addresses. It will probably be much quicker, however, to lease these addresses from an ISP that has already reserved them from ICANN.

Alternatively, if the network is behind a firewall, administrators can make up their own IP addressing scheme without adhering to ICANN standards. A **firewall** is a special kind of router that secures a network from outside penetration via the Internet; it is commonly

used to protect businesses with a presence on the Web. (You will learn more about firewalls and other network security measures in Chapter 15.) For example, you could use a firewall to protect the Jan's Jams network from security breaches related to e-commerce transactions on its Web site. A firewall isolates the network from the Internet at large. As a result, valid IP addresses aren't required within the network. If your office machines aren't really using valid IP addresses, however, how will your staff get through the firewall and onto the Internet? When staff members request access to machines outside your office LAN, they must be assigned valid Internet IP addresses at the firewall.

Isolating a network behind a firewall and then using your own address scheme provide useful management benefits. (For example, if you ran a large LAN, you could assign all machines on the third floor of an office building addresses beginning with 10.3.) In addition, this scheme allows an organization to use more IP addresses than it could if it assigned ICANN-sanctioned numbers to each machine.

A secondary number, known as a subnet mask, is also assigned as part of the TCP/IP configuration process. A subnet mask allows large networks to be subdivided into smaller subnetworks known as subnets. The subnet mask identifies to the network software which addresses appear on the same local network and which addresses need to be contacted through a router. Subnetting is a complex, but highly useful aspect of TCP/IP networking. Chapter 11 explains subnetting in more detail.



Recall from Chapter 1 that a host is any machine on a network that enables resource sharing. All individual computers connected through a TCP/IP network can be called **hosts**. This idea represents a slightly different interpretation of the term "host," because probably not all computers on a TCP/IP network will facilitate resource sharing (though theoretically, they could).

IP address data are sent across the network in binary form, with each of the four octets consisting of eight bits. For example, the IP address 131.127.3.22 is the same as the binary number 10000011 01111111 00000011 00010110. Converting from the dotted decimal notation to binary number is a simple process when you use a scientific calculator, such as the one available with the Windows 2000 operating system.

To convert the first octet (131) of the IP address above to a binary number:

1. On a Windows 2000 computer, click **Start**, point to **Programs**, point to **Accessories**, then click **Calculator**.
2. Click **View**, then click **Scientific**. Make sure that the **Dec option button** is selected.
3. Type **131**, then click the **Bin option button**. The binary equivalent of the number 131, 10000011, appears in the display window.



You can reverse this process to convert a binary number to a decimal number.

Every host on a network must have a unique number, as duplicate addresses will cause problems on a network. If you add a host to a network and its IP address is already in use by another host on the subnet, an error message will be generated on the new client and its TCP/IP services will be disabled. The existing host may also receive an error message, but can continue to function normally.

You can assign IP addresses manually, by modifying the client workstation's TCP/IP properties. A manually assigned IP address is called a **static IP address** because it does not change automatically. It changes only when you reconfigure the client's TCP/IP properties. Alternatively, you can have IP addresses assigned automatically through the **Dynamic Host Configuration Protocol (DHCP)**, an Application layer protocol in the TCP/IP suite. Most networks provide the capability of dynamically assigning IP addresses.

You must take care to avoid assigning duplicate addresses. For example, suppose you spend an afternoon manually assigning IP addresses to 50 Windows 2000 Professional machines in a computer lab. After the forty-eighth machine, you feel tired and mistakenly give the same IP address, 198.5.77.207, to machines 49 and 50. The next day, a student uses computer 49 to pick up her e-mail. A few minutes later, a student turns on computer 50. When he tries to connect to the network, he receives an error message effectively saying that "IP address 198.5.77.207 is being used by 08-AF-82-01-44-CE," where 08-AF-82-01-44-CE is the MAC address of computer 49. The student at computer 50 cannot proceed until either computer 49 is shut down or changes its IP address or until he changes the IP address of computer 50.

Using a DHCP server to assign IP addresses can almost completely eliminate duplicate-addressing problems. (DHCP is described in detail in Chapter 11.) You can envision DHCP as a kind of resource manager for IP addresses. To understand how it works, think of how a health club attendant might hand out towels. When you arrive at the club, the attendant at the desk hands you a towel. You don't care which towel it is, because all towels are the same. You use the towel while you're at the club, then return it when you no longer need it. While you have possession of the towel, no one else can use it. Once you return the towel, it will be returned to the group of towels that the attendant might hand out to other health club members.

Both Windows 2000 and Windows 98 workstations allow users to view their current IP addresses. To view your current IP information on a Windows 98 workstation connected to a network:

1. Click **Start**, then click **Run**. The Run dialog box opens.
2. Type **winipcfg** in the Open text box.
3. Click **OK**. An IP Configuration window containing four numbers appears. The IP address appears second in the list of numbers.
4. To view more information about your network addressing, click the **More Info** button at the lower-right corner of the IP Configuration window. A larger IP

Configuration window appears, as shown in Figure 3-4. Some examples of additional information you can find are the host name, DNS server and DHCP server address.

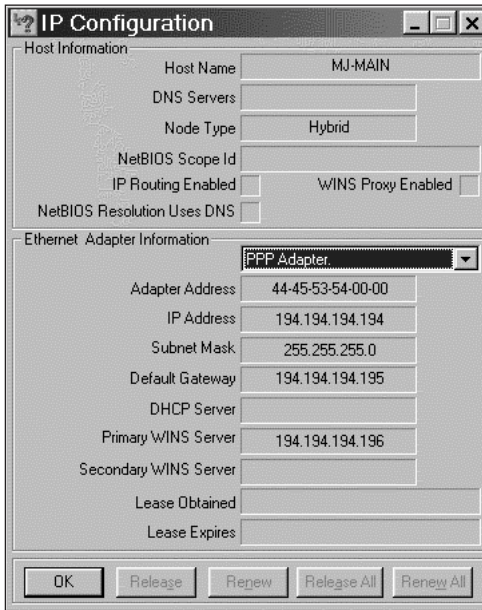


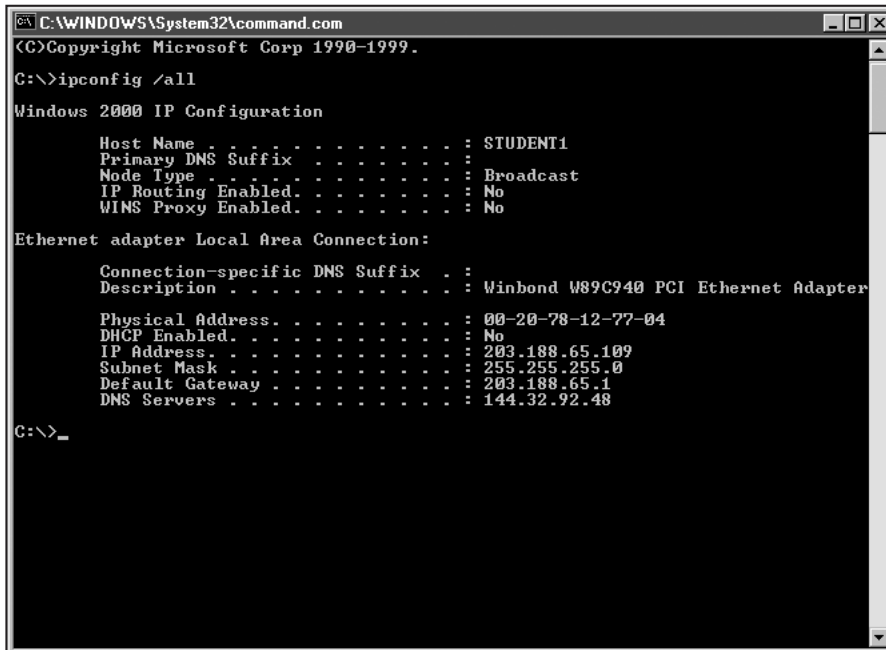
Figure 3-4 An example of an IP Configuration window

5. Click **OK** to close the IP Configuration window.

To view your current IP address from a workstation running Windows 2000:

1. Click **Start**, then click **Run**. The Run dialog box opens.
2. In the Open text box, type **command**, then click **OK**. The Command Prompt window opens.
3. At the DOS prompt, type **ipconfig /all**. Your workstation's IP address information is displayed, similar to the information shown in Figure 3-5.

In addition to using IP addresses, TCP/IP networks use names for networks and hosts, so as to make them more easily identifiable to humans. Each host (computer on a TCP/IP network) requires a host name. Each network must have a network name, also known as a **domain name**. If, while viewing the IP configuration window on your Windows 98 workstation, you click the More Info button, you would see your workstation's host name and domain name in the very first field at the top of the IP configuration window, under the Host Information section.



```

C:\WINDOWS\System32\command.com
<C>Copyright Microsoft Corp 1990-1999.
C:\>ipconfig /all

Windows 2000 IP Configuration

    Host Name . . . . . : STUDENT1
    Primary DNS Suffix . . . . . :
    Node Type . . . . . : Broadcast
    IP Routing Enabled . . . . . : No
    WINS Proxy Enabled . . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . :
    Description . . . . . : Winbond W89C940 PCI Ethernet Adapter
    Physical Address . . . . . : 00-20-78-12-77-04
    DHCP Enabled. . . . . : No
    IP Address . . . . . : 203.188.65.109
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 203.188.65.1
    DNS Servers . . . . . : 144.32.92.48

C:\>_

```

Figure 3-5 IP address information on a Windows 2000 workstation

Together, the host name and domain name constitute the **fully qualified domain name (FQDN)**. For example, your host name might be student1 while your domain name is sacc.tec.ca.us. Therefore, your fully qualified domain name would be student1.sacc.tec.ca.us. Other users on a TCP/IP network, such as the Internet, would identify you by this name, and other machines would associate your IP address with this name. Your host name, “student1,” is a field the network administrator configures on the computer (see the Installing Protocols section later in this chapter). The rest of the name, “sacc.tec.ca.us,” is the network’s domain name. Domain names must follow strict rules and depend on a domain name server to resolve network addresses with the domain name assigned to them.

Chapter 11 covers TCP/IP naming services in more detail. For now, it is enough to know that every node on a TCP/IP network requires a unique host name plus a domain name to communicate over the Internet.

IPX/SPX

Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX) is a protocol originally developed by Xerox, then modified and adopted by Novell in the 1980s for its NetWare network operating system. IPX/SPX is required to ensure the interoperability of LANs running NetWare versions 3.2 and lower and can be used with LANs running higher versions of the NetWare operating system. Other network operating systems, such as Windows 2000, and workstation operating systems, such as

Windows 98, can use IPX/SPX to internetwork with Novell NetWare systems. In the Windows 2000 network operating system, IPX/SPX is called NWLink.

IPX/SPX, like TCP/IP, is a combination of protocols that reside at different layers of the OSI Model. Also like TCP/IP, IPX/SPX carries network addressing information, so it is routable.

IPX/SPX Compared to the OSI Model

IPX/SPX contains a number of subprotocols that belong to different layers of the OSI Model. It does not contain as many subprotocols as TCP/IP, however. For this reason, it is not typically assigned its own model of communications. The IPX/SPX subprotocols roughly correspond to the OSI Model as shown in Figure 3-6. Notice that IPX corresponds to the Network layer of the OSI Model and SPX corresponds to the Transport layer. Later in this chapter, you will be introduced to the higher-level IPX/SPX protocols, including NCP, SAP, and RIP.

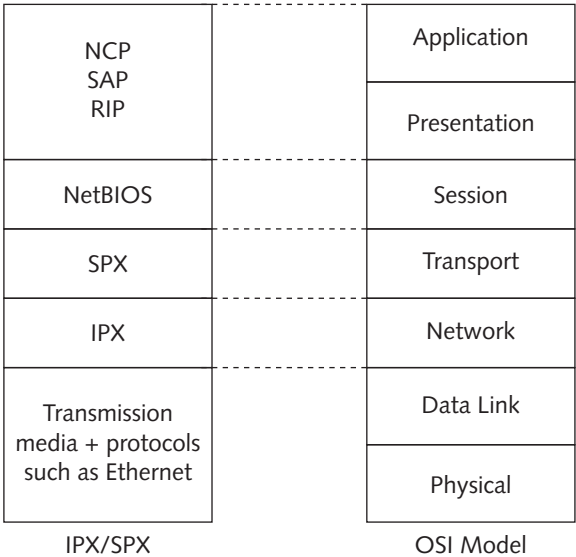


Figure 3-6 IPX/SPX compared to the OSI Model

IPX/SPX Core Protocols

The core protocols of IPX/SPX provide services at the Transport and Network layers of the OSI Model. As you might guess, the most important subprotocols are IPX and SPX. These and other core protocols are explained following.

IPX

Internetwork Packet Exchange (IPX) operates at the Network layer of the OSI Model and provides routing and internetwork services, similar to IP in the TCP/IP suite. Like IP, IPX also uses datagrams to transport data. IPX is a connectionless service because

it does not require a session to be established before it transmits, and it does not guarantee that data will be delivered in sequence or without errors. In summary, it is an efficient subprotocol with limited capabilities. All IPX/SPX communication relies upon IPX, however, and upper-layer protocols handle the functions that IPX cannot perform. The elements of an IPX datagram are described in the following list, and its structure is illustrated in Figure 3-7.

- *Checksum*—Provides integrity checking for the IPX datagram, or packet.
- *Packet length*—Identifies the length of the complete IPX packet in bytes.
- *Transport control*—Tracks the number of routers that a packet has passed through (similar to the TTL parameter in IP). IPX/SPX packets are discarded by the sixteenth router they encounter.
- *Packet type*—Defines the service offered or required by the packet.
- *Destination Network*—Indicates the network address of the destination network.
- *Destination node address*—Indicates the node address of the destination node (that is, its MAC address).
- *Destination socket*—Refers to the process address on the destination node. A **socket** is a logical address assigned to a specific process running on a computer. Some sockets are reserved for operating system functions.
- *Source network*—Indicates the network address of the source network.
- *Source node address*—Indicates the node address of the source node, equivalent to its MAC address.
- *Source socket*—Indicates the socket address of the process running on the source node.
- *Data*—Contains data originally sent by the source and the SPX packet.

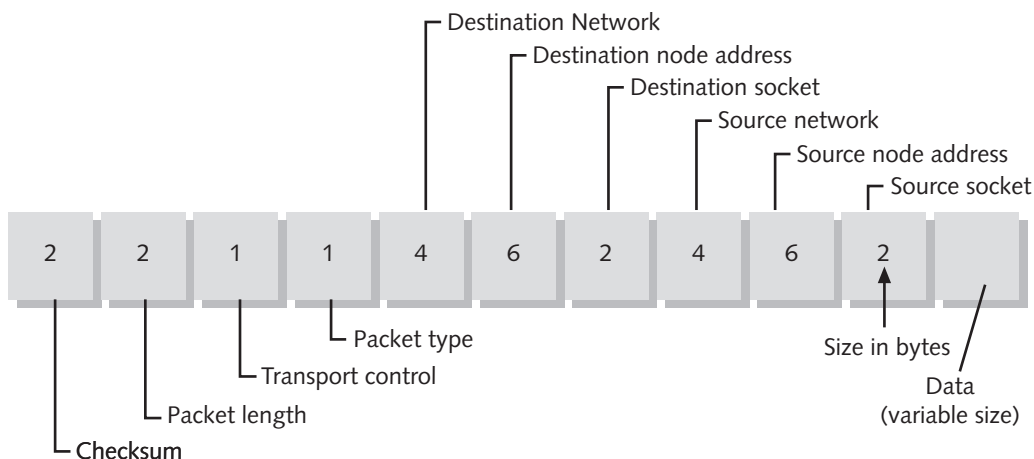


Figure 3-7 Components of an IPX datagram

SPX

Sequenced Packet Exchange (SPX) belongs to the Transport layer of the OSI Model. It works in tandem with IPX to ensure that data are received whole, in sequence, and error free. SPX, like TCP in the TCP/IP suite, is a connection-oriented protocol and therefore must verify that a session has been established with the destination node before it will transmit data. It can detect whether a packet was not received in its entirety. If it discovers a packet has been lost or corrupted, SPX will resend the packet.

The SPX information is enveloped by IPX. That is, its fields sit inside the data field of the IPX datagram, as depicted in Figure 3-8.

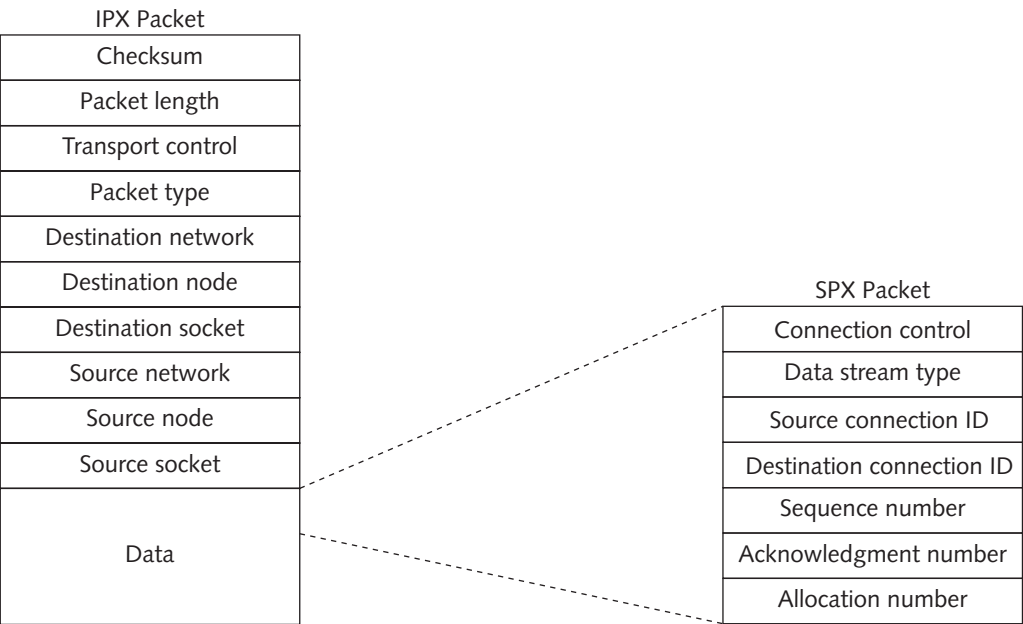


Figure 3-8 SPX packet encapsulated by an IPX packet

The SPX packet, like the TCP segment, contains a number of fields to ensure data reliability. An SPX packet consists of a 42-byte header followed by 0 to 534 bytes of data. An SPX packet can be as small as 42 bytes (the size of its header) or as large as 576 bytes. The following list describes each field in an SPX packet and its function:

- *Connection control*—Indicates whether the packet is a system or application packet.
- *Data stream type*—Indicates the type of data found in the packet—for example, whether the packet is the beginning or the end of a data stream.
- *Source connection ID*—Identifies the source node.

- *Destination connection ID*—Identifies the destination node.
- *Sequence number*—Indicates the number of packets exchanged in one direction on the connection.
- *Acknowledgment number*—Identifies the sequence number of the next packet that an SPX connection expects to receive.
- *Allocation number*—Used to manage flow control between communicating applications.

Service Advertising Protocol (SAP)

The **Service Advertising Protocol (SAP)** works in the Application, Presentation, and Session layers of the OSI Model and runs directly over IPX. NetWare servers and routers use SAP to advertise to the entire network which services they can provide. For example, a server that functions as a print server might use SAP to effectively announce to every node on the network, “I’m available to help you print.” By default, SAP broadcasts occur every 60 seconds. Because SAP uses the broadcast mode to transmit its information, it may generate a great deal of unnecessary traffic on the network, slowing down other, more important transmissions. One way to reduce this traffic is to increase the time between SAP broadcasts from 60 seconds to a few minutes.

Once devices have advertised their availability through SAP, SAP servers maintain a database of device names correlated with their IPX addresses. When a client needs to request a service from a particular device, the client queries the SAP database and the database then provides the IPX address for the desired device. In this way, the protocol frees users from having to know the IPX addresses of other servers and workstations on their network.

On networks that use NetWare Directory Services (NDS), which is discussed in detail in Chapter 9, SAP may not be necessary because NDS will point clients to the necessary service. For example, rather than having a server advertise through SAP every 60 seconds that it can perform printer services, NDS can point clients directly to that server when the client needs to print.

NetWare Core Protocol (NCP)

The **NetWare Core Protocol (NCP)** handles requests for services, such as printing and file access, between clients and servers. NCP works over IPX and within the Presentation and Session layers of the OSI Model. In essence, NCP acts as a translator between the workstation’s operating system and the NetWare operating system. It uses a request-and-response mechanism to accomplish its translation; that is, once a client asks it to request a service, it notifies the server that a request is pending. NCP then waits for the server to acknowledge the request before it allows the workstation to transmit data. Although this exchange results in high reliability, it also generates extra traffic and may add to congestion on networks, such as WANs, that use routers.

Addressing in IPX/SPX

Maintaining network addresses for clients running IPX/SPX is easier than maintaining addresses for TCP/IP networks, because IPX/SPX networks primarily rely on the MAC address for each workstation (although addressing for IPX/SPX servers can be somewhat more complex). Just as with TCP/IP networks, IPX/SPX networks require that each node on a network be assigned a unique address to avoid communication conflicts. Because IPX is the component of the protocol that handles addressing, addresses on an IPX/SPX network are called **IPX addresses**. IPX addresses contain two parts: the network address (also known as the **external network number**) and the node address.

The network administrator establishes a network address when installing the NetWare operating system software on a server. The network address must be an 8-bit hexadecimal address, which means that each of its bits can have a value of either 0–9 or A–F. An example of a valid network address is 000008A2. The network address then becomes the first part of the IPX address on all nodes that use the particular server as their primary server.



The address 00000000 is a null value and cannot be used as a network address. The address FFFFFFFF is a broadcast address and also cannot be assigned as a network address.

The second part of an IPX address, the node address, is equal to the network device's MAC address. Because every network interface card should have a unique MAC address, no possibility of duplicating IPX addresses exists under this system. In addition, the use of MAC addresses means that you need not configure addresses for the IPX/SPX protocol on each client workstation. Instead, they are already defined by the NIC. Adding a MAC address to the network address example used previously, a complete IPX address for a workstation on the network might be 000008A2:0060973E97F3.

Imagine you are the administrator for a building's NetWare 3.11 network with one server and 40 connected workstations, plus five printers. Your network is connected to six other networks on a large corporate campus. A colleague alerts you that one of the Accounting department's four workstations is generating excessive error messages. You need to determine the malfunctioning workstation's IPX address before you can disconnect it from the server. Because you installed the network originally, you know that your network address is 0000AAAA. The workstation's address must therefore begin with 0000AAAA (addresses for workstations coming from networks elsewhere on campus will begin with a different sequence). You also know that the Accounting department's computers contain NICs manufactured by Compaq. You look up the manufacturer's Ethernet code (the first part of the MAC address) and find that it is 00805F. In the list of currently attached workstations, you find only one IPX address that matches the pattern beginning with 0000AAAA:00805F—a machine with the full address of 0000AAAA:00805F059822. You can correctly assume that it is the faulty Accounting workstation.

In addition to the network and node addresses, processes running on IPX-enabled workstations are identified by socket addresses. When a process needs to communicate on the network, it requests that a socket number be assigned to it. Any packets addressed to that socket are passed on to the corresponding process. This approach enables nodes to route communications between their own sockets. An example of a socket address is 456h; Novell has reserved this particular socket for its diagnostics process. Socket addresses are appended to IPX addresses, so an example of a complete IPX address for a socket would be 000008A2:0060973E97F3:456h.

To view your Windows 98 or Windows 2000 workstation's IPX address while connected to a NetWare server running version 4.0 or higher:

1. Click **Start**, then click **Run**. The Run dialog box opens.
2. In the Open text box, type **command**, then click **OK**. The Command Prompt window opens.
3. Change directories to a drive letter you have mapped to the network (for example, typing the command F: will work on most networks.)
4. At the DOS prompt, type **nlist user-XXXXXX /a** where "XXXXXX" is your NetWare logon ID. (The nlist command in NetWare is a listing command, while user defines the kind of information that you want to list and the /a parameter indicates that you want to see the address for the specified user.) As a result of this command, you see the user ID you specified along with its corresponding IPX address.

To view your Windows 98 or Windows 2000 workstation's IPX address while connected to a NetWare server running a version lower than 4.0:

1. Click **Start**, then click **Run**. The Run dialog box opens.
2. In the Open text box, type **command**, then click **OK**. The Command Prompt window opens.
3. At the DOS prompt, type **userlist user=XXXXXX /a** where XXXXX is your NetWare logon ID. In NetWare versions lower than 4.0, the userlist command performs the same function as the nlist command in NetWare versions 4.0 and higher. You see the user ID you specified along with its corresponding IPX address.

NETBIOS AND NETBEUI

NetBIOS (Network Basic Input Output System) is a protocol originally designed for IBM to provide Transport and Session layer services for applications running on small, homogenous networks. Microsoft adopted IBM's NetBIOS as its foundation protocol, initially for networks using LAN Manager or Windows for Workgroups, but then added an Application layer component on top of NetBIOS called the **NetBIOS**

Enhanced User Interface (NetBEUI; pronounced, “net-bóo-ee”). NetBEUI is a fast and efficient protocol that consumes few network resources, provides excellent error correction, and requires little configuration. It can support only 254 connections, however, and does not allow for good security. Furthermore, because NetBEUI uses Data Link layer addressing rather than Network layer addressing, it is not routable. (If you want to use NetBIOS in a routable fashion, instead of using NetBEUI, you can use a routable protocol such as TCP/IP to encapsulate NetBIOS. The preferred method, however, is to migrate a NetBEUI network to a network relying entirely on TCP/IP.) Thus, this protocol is not suitable for large networks. Today NetBEUI is most commonly used in small Microsoft-based networks to integrate legacy, peer-to-peer networks. In newer Microsoft-based networks, TCP/IP has become the protocol of choice because it is more flexible and scalable than NetBEUI.

NetBIOS and NetBEUI Compared to the OSI Model

Because neither NetBIOS nor NetBEUI provides services at all layers of the OSI Model, both are commonly paired with other protocol suites, such as IPX/SPX or TCP/IP when placed in the OSI Model. Figure 3-9 shows how NetBIOS and NetBEUI fit into the OSI Model.

Application		
Presentation		
Session		NetBIOS
Transport		NetBEUI
Network		
Data Link		
Physical		

Figure 3-9 NetBIOS/NetBEUI compared to the OSI Model

NetBIOS Addressing

You have learned that NetBIOS does not contain a Network layer and therefore cannot be routed. To transmit data between network nodes, however, NetBIOS needs to reach each workstation. For this reason, network administrators must assign a NetBIOS name to each workstation. The NetBIOS name can consist of any combination of 16 or fewer alphanumeric characters (the only exception is that you cannot begin a NetBIOS name with an asterisk). Once NetBIOS has found a workstation's NetBIOS name, it will discover the workstation's MAC address and then use this address in further communications

with the workstation. For example, a valid NetBIOS name is MY_COMPUTER. You might use NetBIOS names when troubleshooting problems on a NetBIOS network.



On networks running both TCP/IP and NetBIOS, it is simplest to make the NetBIOS name identical to the TCP/IP host name.

To view the NetBIOS name of a computer running the Windows 2000 operating system:

1. Click **Start**, point to **Settings**, then click **Control Panel**. The Control Panel window opens.
2. Double-click the **System** Icon. The System Properties dialog box opens.
3. Click the **Network Identification** tab. As shown in Figure 3-10, the first item in the Identification tab is the full computer name. The full computer name is the same as the workstation's NetBIOS name.

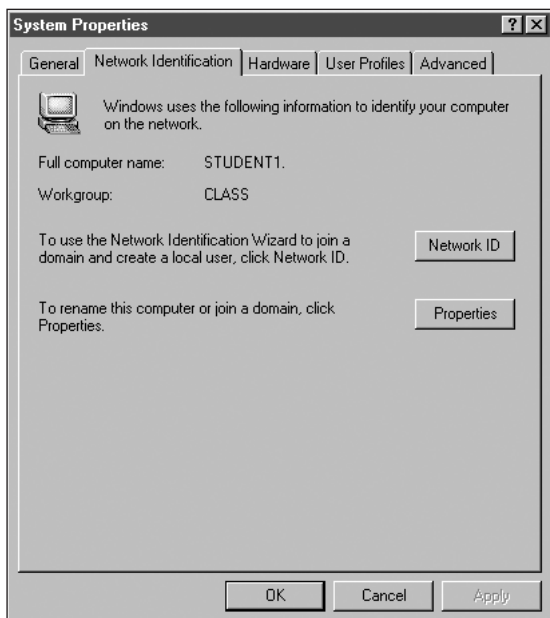


Figure 3-10 Network Identification tab in the System Properties dialog box

APPLETALK

Businesses and institutions involved in art or education, such as advertising agencies, elementary schools, and graphic designers, often use Apple Macintosh computers. **AppleTalk** is the protocol suite used to interconnect Macintosh computers. Although AppleTalk was originally designed to support peer-to-peer networking among

Macintoshes, it can now be routed between network segments and integrated with NetWare- or Microsoft-based networks.

An AppleTalk network is separated into logical groups of computers called **AppleTalk zones**. Each network can contain multiple zones, but each node can belong to only one zone. AppleTalk zones enable users to share file and printer resources on one another's Macintoshes. Zone names are not subject to the same strict naming conventions that TCP/IP and IPX/SPX networks must follow. Instead, zone names typically describe a department or other group of users who share files. An example of a zone name is "Sales and Marketing."

Although Apple has improved AppleTalk's ability to use different network models and span network segments, it remains unsuited to large LANs or WANs. Even Apple has begun supporting the TCP/IP protocol to integrate Macintoshes with other networks, including the Internet.

AppleTalk Compared to the OSI Model

AppleTalk is a complete protocol suite containing services that fit into each layer of the OSI Model, as depicted in Figure 3-11.

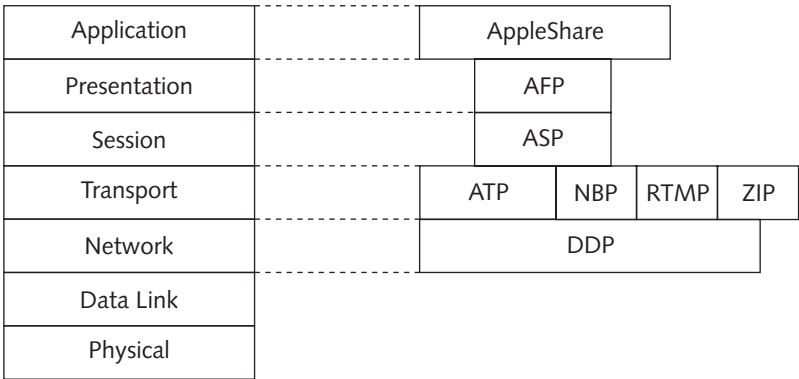


Figure 3-11 The AppleTalk protocol compared to the OSI Model

The AppleTalk subprotocols that are significant for node-to-node communications are described in the following list:

- *AppleShare*—Provides file sharing services, print queueing services, password access to files or folders, and user accounting information.
- *AppleTalk Filing Protocol (AFP)*—Provides transparent access to files on both local and remote systems.
- *AppleTalk Session Protocol (ASP)*—Establishes and maintains connections between nodes and servers.

- *AppleTalk Transaction Protocol (ATP)*—Ensures reliable delivery of data by checking connections between nodes, checking packet sequence, and retransmitting any data packets that become lost.
- *Name Binding Protocol (NBP)*—Translates human-readable node names into numeric AppleTalk addresses.
- *Routing Table Maintenance Protocol (RTMP)*—Maintains a routing table of AppleTalk zones and their networks, and uses ZIP to manage data in the routing table.
- *Zone Information Protocol (ZIP)*—Updates zone information maps that tie zones to their networks for routing purposes.
- *Datagram Delivery Protocol (DDP)*—Assigns an AppleTalk node's address upon start-up and manages addressing for communications between AppleTalk nodes.

Addressing in AppleTalk

You have learned that AppleTalk uses zones and that zone names can be plain words or numbers with no restrictions. In addition to zone names, AppleTalk uses node IDs and network numbers to identify computers on a network.

An **AppleTalk node ID** is a unique 8-bit or 16-bit number that identifies a computer on an AppleTalk network. AppleTalk assigns a node ID to each workstation when the workstation first connects to the network. The ID is randomly chosen from a group of currently available addresses. Once a device has obtained an address, it stores it for later use.

An **AppleTalk network number** is a unique 16-bit number that identifies the network to which a node is connected. Its use allows nodes from several different networks to communicate.

AppleTalk addressing is simple because it allows you to identify a group of shared addresses from the server. When clients attach to that server they pick up an address, thus eliminating the need to configure addresses on each separate workstation.

INSTALLING PROTOCOLS

The protocols you install will depend on which operating system you are running. This section describes how to install the most commonly used protocols on Windows 98 and Windows 2000 client workstations. Chapters 8 and 9 discuss installing and configuring protocols on the two of the most commonly used network operating systems, NetWare 5.x and Windows 2000.

Installation is merely the first step in making protocols work. After they are installed, you must bind them to the NICs and services they will run on or with. **Binding** is the process of assigning one network component to work with another. Once you install a protocol on a Windows 2000 or Windows 98 workstation, it binds itself automatically to the NICs

and services it finds on the computer. However, depending on the computer's version of Windows, you may have to restart the machine for the bindings to take effect. For optimal network performance, you should install and bind only those protocols that you absolutely need. For example, a Windows 2000 server will attempt to use bound protocols in the order in which they appear in the protocol listing until it finds the correct one for the response at hand. This approach wastes processing time, making it more efficient to bind only the protocols you need.

Installing Protocols on a Windows 2000 Professional Workstation

The following exercise shows you how to install the NetBEUI protocol on a Windows 2000 Professional workstation (note that TCP/IP would normally be already installed with the operating system, while NetBEUI would not). The process of installing other protocols on a Windows 2000 Professional workstation is identical.

1. Log on to the workstation as an Administrator.
2. Click **Start**, point to **Settings**, then click **Network and Dial-up Connections**.
3. Right-click the **Local Area Connection** icon and click **Properties** in the shortcut menu. The Properties dialog box opens.
4. Click **Install**. The Select Network Component Type dialog box opens.
5. Click **Protocol** in the list of Network Component Types.
6. Click **Add**. The Select Network Protocol dialog box opens.
7. In the list of network protocols, click **NetBEUI Protocol**, then click **OK**.
Notice that NetBEUI now appears in the list of network components.
8. Click **Close**. The Local Area Connection Properties dialog box closes and your change is saved.
9. To verify that the protocol was installed, click **Start**, point to **Settings**, then click **Network and Dial-up Connections**.
10. Right-click the **Local Area Connection** icon and click **Properties**. The Properties dialog box appears.
11. Verify that NetBEUI Protocol appears in the list of installed protocols.
12. Click **Cancel** to close the Network dialog box.

On a Windows 2000 workstation, you can install any other protocol in the same manner as you installed the NetBEUI protocol. Although the NetBEUI protocol requires no further configuration, usually you must configure the AppleTalk and TCP/IP protocols after installing them. Chapter 11 covers TCP/IP configuration in detail.

Installing Protocols on a Windows 98 Workstation

The following exercise shows you how to install the TCP/IP protocol on a Windows 98 workstation (it assumes that you have either previously removed the TCP/IP protocol or are installing it again; Windows 98 installations include the TCP/IP protocol by default).

1. Right-click the **Network Neighborhood** icon, then click **Properties**.
2. Verify that the **Configuration** tab is selected.
3. Click **Add**. The Select Network Component Type window opens.
4. Double-click **Protocol**. The Select Network Protocol window opens.
5. In the list of manufacturers, click **Microsoft**.
6. In the list of protocols, click **TCP/IP**.
7. Click **OK**, and then click **OK** again.
8. If TCP/IP is not already installed on your workstation, you will be prompted to restart your workstation to allow the changes to take effect. Click **Yes** to restart your workstation.
9. To verify that the protocol was installed, right-click the **Network Neighborhood** icon, then click **Properties**.
10. Verify that the **Configuration** tab is selected. One of the items in the list of services, clients, and protocols should be TCP/IP.
11. Click **Cancel** to close the Network properties window.

You can add other protocols to your Windows 98 workstation in the same manner. Although usually you do not need to configure IPX/SPX or NetBEUI after installation, you must configure TCP/IP unless you are using DHCP. Chapter 11 covers TCP/IP configuration in detail.

It is possible to bind multiple protocols to the same network adapter. In fact, this is necessary on networks that use more than one type of protocol. In addition, a workstation may have multiple NICs, in which case several different protocols might be bound to each NIC. What's more, the same protocol may be configured differently on different NICs. For example, let's say you managed a NetWare server that contained two NICs and provided both TCP/IP and IPX/SPX communications to many clients. After installing the TCP/IP protocol on the server, you would need to configure TCP/IP separately for each NIC using the network operating system's protocol configuration utility. Similarly, you would need to configure IPX/SPX separately for each NIC. If you did not configure the protocols for each NIC separately, clients would not know which NIC to address when sending and receiving information to and from the server.

CHAPTER SUMMARY

- ❑ Protocols define the standards for communication between nodes on a network. The term *protocol*, in networking, can refer to a group, or suite, of individual protocols that work together to accomplish data translation, data handling, error checking, and addressing.
- ❑ Protocols vary by speed, transmission efficiency, utilization of resources, ease of setup, compatibility, and ability to travel between one LAN segment and another. Protocols that can span more than one LAN segment are routable, because they carry Network layer and addressing information that can be interpreted by a router.
- ❑ The most commonly used protocols are TCP/IP, IPX/SPX, NetBIOS, and AppleTalk. You may also find other, outdated protocols in use, such as SNA and DLC.
- ❑ A network that uses more than one protocol is called a multiprotocol network. Multiprotocol networks are common in businesses with well-established LANs that have evolved from a legacy system to a newer, more efficient one.
- ❑ TCP/IP is fast becoming the most popular network protocol because of its low cost and its ability to communicate between a multitude of dissimilar platforms. It is a de facto standard on the Internet and is commonly the protocol of choice on LANs. TCP/IP is routable and flexible.
- ❑ The TCP/IP suite of protocols can be divided into four layers that roughly correspond to the seven layers of the OSI Model: the Application layer, the Transport layer, the Internet layer, and the Network Interface layer.
- ❑ The TCP/IP core protocols operate in the Transport or Network layers of the OSI Model, where they provide communications between hosts on a network. The most significant core protocols in the TCP/IP suite are IP and TCP.
- ❑ The Internet Protocol (IP) belongs to the Internet layer of the TCP/IP Model and provides information about how and where data should be delivered. IP is the subprotocol that enables TCP/IP to internetwork—that is, to traverse more than one LAN segment and more than one type of network through a router.
- ❑ The Transmission Control Protocol (TCP) belongs to the Transport layer of the TCP/IP suite and provides reliable data delivery services. TCP is a connection-oriented subprotocol, which means that it requires a connection to be established between communicating nodes before it will transmit data. TCP sits on top of the IP subprotocol and compensates for IP's reliability deficiencies with its checksum, flow control, and sequencing information.
- ❑ The User Datagram Protocol (UDP), like TCP, sits in the Transport layer, between the Internet layer and the Application layer of the TCP/IP Model. Unlike TCP, however, UDP is a connectionless transport service. It offers no error checking and no assurance that packets will be received in the correct sequence. UDP's lack of sophistication actually makes it more efficient than TCP and useful in situations where data must be transferred quickly, such as live audio or video transmissions over the Internet.

- Internet Control Message Protocol (ICMP), another TCP/IP core protocol, notifies the sender that something has gone wrong in the transmission process and that packets were not delivered. ICMP sits between IP and TCP in the Internet layer of the TCP/IP Model and reports which networks are unreachable and which packets have been discarded because the allotted time for their delivery has expired.
- The Address Resolution Protocol (ARP) belongs to the Internet layer of the TCP/IP Model. It obtains the MAC (physical) address of a host, or node, then creates a local database that maps the MAC address to the host's IP (logical) address.
- The TCP/IP suite includes a number of useful Application layer protocols, such as Telnet, FTP, SMTP, and SNMP.
- Each IP address is a unique 32-bit number, divided into four octets that are separated by periods. An example of a valid IP address is 144.92.43.178. An IP address contains two types of information: network and host.
- All nodes on a Class A network share the first octet of their IP numbers, a number between 1 and 126. Nodes on a Class B network share the first two octets, and all their IP addresses begin with a number between 128 and 191. Class C network IP numbers share the first three octets, with their first octet being a number between 192 and 223.
- The range of addresses beginning with 127 is reserved for loopback information. The IP address 127.0.0.1 is called a loopback address. When you try to contact this IP number, you actually communicate with your own machine. This address is useful for troubleshooting problems with a workstation's TCP/IP communications.
- Every host on a network must have a unique number, as duplicate addresses will cause problems. If a host is added to a network and its IP address is already assigned to another host on the subnet, an error message will be generated on the new client and its TCP/IP services will be disabled. The existing host may also receive an error message, but can continue to function normally.
- Although you may assign IP addresses manually, you must take care to avoid assigning duplicate addresses. IP addresses assigned manually are called static IP addresses. Most networks provide the capability of dynamically assigning IP addresses through the Dynamic Host Configuration Protocol (DHCP) protocol, an Application layer protocol in the TCP/IP suite. Using a DHCP server to assign IP addresses can nearly eliminate duplicate-addressing problems.
- Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX) is a protocol originally developed by Xerox, then modified and adopted by Novell in the 1980s for its NetWare network operating system. IPX/SPX is required for interoperability with LANs running NetWare versions 3.2 and lower; it can also be used with LANs running higher versions of the NetWare operating system. IPX/SPX, like TCP/IP, is a suite of protocols that reside at different layers of the OSI Model. Also like TCP/IP, IPX/SPX carries network addressing information, so it is routable.

- The core protocols of IPX/SPX provide services at the Transport and Network layers of the OSI Model. Its most important subprotocols are IPX and SPX.
- Internetwork Packet Exchange (IPX) operates at the Network layer of the OSI Model and provides routing and internetwork services, similar to IP in the TCP/IP suite. IPX uses datagrams to transport data. This protocol is a connectionless service because it does not require a session to be established before it transmits data, and it does not guarantee that data will be delivered in sequence or without errors. It is an efficient subprotocol with limited capabilities.
- Sequenced Packet Exchange (SPX) belongs to the Transport layer of the OSI Model. It works in tandem with IPX to ensure that data are received whole, in sequence, and error free. SPX is a connection-oriented protocol and therefore must verify that a session has been established with the destination node before it will transmit data. It can detect whether a packet was not received in its entirety; if it discovers that a packet has been lost or corrupted, SPX will resend the packet.
- The Service Advertising Protocol (SAP) works in the Application, Presentation, Session, and Transport layers of the OSI Model and runs directly over IPX. NetWare servers and routers use SAP to advertise to the entire network which services they can provide.
- The NetWare Core Protocol (NCP) handles requests for services, such as printing and file access, between clients and servers. NCP works over IPX and within the Presentation and Session layers of the OSI Model. It acts as a translator between the workstation's operating system and the NetWare operating system.
- Because IPX is the component of the protocol that handles addressing, addresses on an IPX/SPX network are called IPX addresses. IPX addresses contain two parts: the network address and the node address. The network address must be an 8-bit hexadecimal address, which means that each of its bits can have a value of either 0–9 or A–F. The second part of an IPX address, the node address, is equal to the network device's MAC address.
- NetBIOS (Network Basic Input Output System) is a protocol originally designed for IBM to provide Transport and Session layer services for applications running on small, homogenous networks.
- Microsoft adopted IBM's NetBIOS as its foundation protocol, initially for networks using LAN Manager or Windows for Workgroups, but then added an Application layer component on top of NetBIOS called the NetBIOS Enhanced User Interface (NetBEUI). NetBEUI is a fast and efficient protocol that consumes few network resources, provides excellent error correction, and requires little configuration. It can support only 254 connections, however, and does not allow for good security. Furthermore, because NetBEUI lacks a Network layer, it is not routable and therefore not suitable for large networks.
- To transmit data between network nodes, NetBIOS needs to know how to reach each workstation. For this reason, network administrators must assign a NetBIOS

name to each workstation. The NetBIOS name can be any combination of 16 or fewer alphanumeric characters (although you cannot begin a NetBIOS name with an asterisk). Once NetBIOS has found a workstation's NetBIOS name, it will discover the workstation's MAC address and then use this address in further communications with the workstation.

- AppleTalk is the protocol suite used to interconnect Macintosh computers. Although AppleTalk was originally designed to support peer-to-peer networking among Macintoshes, it can now be routed between network segments and integrated with NetWare- or Microsoft-based networks.
- An AppleTalk network is separated into logical groups of computers called AppleTalk zones. Each network can contain multiple zones, but each node can belong to only one zone. AppleTalk zones enable users to share file and printer resources on one another's Macintoshes. Zone names typically describe a department or other group of users who share files.
- Although Apple has improved AppleTalk's ability to use different network models and span network segments, it remains unsuited to large LANs or WANs. Even Apple has begun supporting the TCP/IP protocol to integrate Macintoshes with other networks, including the Internet.
- In addition to zone names, AppleTalk uses node IDs and network numbers to identify computers on a network. An AppleTalk node ID is a unique 8- or 16-bit number that identifies a computer on an AppleTalk network. AppleTalk assigns a node ID to each workstation when the workstation connects to the network. An AppleTalk network number is a unique 16-bit number that identifies the network to which a node is connected. AppleTalk addressing can be managed centrally from the server.
- Although some protocols, such as NetBIOS, require no configuration after they are installed, more complex protocols, such as TCP/IP, do require configuration.

KEY TERMS

Address Resolution Protocol (ARP) — A core protocol in the TCP/IP suite that belongs in the Internet layer. It obtains the MAC (physical) address of a host, or node, and then creates a local database that maps the MAC address to the host's IP (logical) address.

AppleTalk — The protocol suite used to interconnect Macintosh computers. Although AppleTalk was originally designed to support peer-to-peer networking among Macintoshes, it can now be routed between network segments and integrated with NetWare- or Microsoft-based networks.

AppleTalk network number — A unique 16-bit number that identifies the network to which an AppleTalk node is connected.

AppleTalk node ID — A unique 8-bit or 16-bit (if you are using extended networking, in which a network can have multiple addresses and support multiple zones) number that identifies a computer on an AppleTalk network.

AppleTalk zone — Logical groups of computers defined on an AppleTalk network.

binding — The process of assigning one network component to work with another.

broadcast — A transmission to all stations on a network.

connection-oriented — A feature of some protocols that requires the establishment of a connection between communicating nodes before the protocol will transmit data.

connectionless — A feature of some protocols that allows the protocol to service a request without requiring a verified session and without guaranteeing delivery of data.

domain name — The symbolic name that identifies a group of IP addresses. Usually, a domain name is associated with a company or other type of organization, such as a university or military unit.

Dynamic Host Configuration Protocol (DHCP) — An Application layer protocol in the TCP/IP suite that manages the dynamic distribution of IP addresses on a network. Using DHCP to assign IP addresses can nearly eliminate duplicate-addressing problems.

external network number — Another term for the network address portion of an IPX/SPX address.

File Transfer Protocol (FTP) — An Application layer protocol used to send and receive files via TCP/IP.

firewall — A specialized device (typically a router, but possibly only a PC running special software) that selectively filters or blocks traffic between networks. A firewall may be strictly hardware-based, or it may involve a combination of hardware and software.

fully qualified domain name (FQDN) — In TCP/IP addressing, the combination of a host and domain name that together uniquely identify a device.

host — A computer connected to a network that uses the TCP/IP protocol.

Internet Control Message Protocol (ICMP) — A core protocol in the TCP/IP suite that notifies the sender that something has gone wrong in the transmission process and that packets were not delivered.

Internet Corporation for Assigned Names and Numbers (ICANN) — The non-profit corporation currently designated by the U.S. government to maintain and assign IP addresses.

Internet Protocol (IP) — A core protocol in the TCP/IP suite that belongs to the Internet layer of the TCP/IP model and provides information about how and where data should be delivered. IP is the subprotocol that enables TCP/IP to internetwork.

internetwork — To traverse more than one LAN segment and more than one type of network through a router.

Internetwork Packet Exchange (IPX) — A core protocol of the IPX/SPX suite that operates at the Network layer of the OSI Model and provides routing and internetwork services, similar to IP in the TCP/IP suite.

Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX) — A protocol originally developed by Xerox, then modified and adopted by Novell in the 1980s for the NetWare network operating system.

IP address — A logical address used in TCP/IP networking. This unique 32-bit number is divided into four groups of octets, or 8-bit bytes, that are separated by periods.

IP datagram — The IP portion of a TCP/IP frame that acts as an envelope for data, holding information necessary for routers to transfer data between subnets.

IPX address — An address assigned to a device on an IPX/SPX network.

loopback address — An IP address reserved for communicating from a node to itself (used mostly for testing purposes). The value of the loopback address is always 127.0.0.1.

multiprotocol network — A network that uses more than one protocol.

NetBIOS Enhanced User Interface (NetBEUI) — Microsoft's adaptation of the IBM NetBIOS protocol. NetBEUI expands on NetBIOS by adding an Application layer component. NetBEUI is a fast and efficient protocol that consumes few network resources, provides excellent error correction and requires little configuration.

NetWare Core Protocol (NCP) — One of the core protocols of the IPX/SPX suite. NCP handles requests for services, such as printing and file access, between clients and servers.

Network Basic Input Output System (NetBIOS) — A protocol designed by IBM to provide Transport and Session layer services for applications running on small, homogeneous networks.

octet — One of the four 8-bit bytes that are separated by periods and together make up an IP address.

port — The address on a host where an application makes itself available to incoming data.

protocol — The rules a network uses to transfer data. Protocols ensure that data is transferred whole, in sequence, and without error from one node on the network to another.

routable — Protocols that can span more than one LAN segment because they carry Network layer and addressing information that can be interpreted by a router.

routing protocols — Protocols that assist routers in efficiently managing information flow.

Sequenced Packet Exchange (SPX) — One of the core protocols in the IPX/SPX suite. SPX belongs to the Transport layer of the OSI Model and works in tandem with IPX to ensure that data are received whole, in sequence, and error free.

Service Advertising Protocol (SAP) — A core protocol in the IPX/SPX suite that works in the Application, Presentation, Session, and Transport layers of the OSI Model and runs directly over IPX. NetWare servers and routers use SAP to advertise to the entire network which services they can provide.

Simple Mail Transfer Protocol (SMTP) — The protocol responsible for moving messages from one e-mail server to another over the Internet and other TCP/IP-based networks.

Simple Network Management Protocol (SNMP) — A communication protocol used to manage devices on a TCP/IP network.

socket — A logical address assigned to a specific process running on a computer. Some sockets are reserved for operating system functions.

static IP address — An IP address that is manually assigned to a device.

subnets — In an internetwork, the individual networks that are joined together by routers.

subprotocols — Small, specialized protocols that work together and belong to a protocol suite.

TCP segment — The portion of a TCP/IP packet that holds TCP data fields and becomes encapsulated by the IP datagram.

TCP/IP core protocols — The subprotocols of the TCP/IP suite.

Telnet — A terminal emulation protocol used to log on to remote hosts using the TCP/IP protocol. Telnet resides in the Application layer of the TCP/IP suite.

Transmission Control Protocol (TCP) — A core protocol of the TCP/IP suite. TCP belongs to the Transport layer and provides reliable data delivery services.

User Datagram Protocol (UDP) — A core protocol in the TCP/IP suite that sits in the Transport layer, between the Internet layer and the Application layer of the TCP/IP model. UDP is a connectionless transport service.

REVIEW QUESTIONS

1. What characteristics make a protocol routable?
 - a. MAC sublayer addresses that can be interpreted by a server
 - b. Network layer and addressing information that can be interpreted by a router
 - c. Logical Link sublayer address information that can be interpreted by a hub
 - d. Transport layer flow control information that can be interpreted by a router
2. Which layer in the TCP/IP model of network communications roughly corresponds to the Physical and Data Link layers of the OSI Model?
 - a. Network Interface layer
 - b. Internet layer
 - c. Transport layer
 - d. Application layer

3. To which layer of the TCP/IP model does the IP protocol belong?
 - a. Network Interface layer
 - b. Internet layer
 - c. Transport layer
 - d. Application layer
4. To which layer of the TCP/IP model does the TCP protocol belong?
 - a. Network Interface layer
 - b. Internet layer
 - c. Transport layer
 - d. Application layer
5. What is the function of ARP?
 - a. to acknowledge that a data frame was received
 - b. to obtain the IP address of a host, then map that IP address to a registered domain name
 - c. to measure the number of dropped packets in a single transmission
 - d. to obtain the MAC address of a host, and then map the MAC address to the host's IP address
6. Which TCP/IP utility might you use to connect to a UNIX host from your PC over the network?
 - a. SNMP
 - b. SMTP
 - c. Telnet
 - d. hup
7. What does SMTP stand for?
 - a. Simple Mail Transfer Protocol
 - b. Simple Message Transport Protocol
 - c. Simple Media Transfer Protocol
 - d. Simple Message Tracking Protocol
8. Which version of IP are most TCP/IP networks currently using?
 - a. 3.0
 - b. 4.0
 - c. 5.0
 - d. 6.0
9. Why might an application be better served by UDP than TCP?
10. An IP address consists of 4 bytes. True or False?

11. Which technique is used to break large TCP/IP-based networks into smaller logical segments?
 - a. subnetting
 - b. subclassing
 - c. reverse lookups
 - d. domain transfers
12. On which Class network would you find the workstation that uses the following IP address: 193.12.176.55?
 - a. A
 - b. B
 - c. C
 - d. D
13. Which of the following is the loopback address?
 - a. 1.1.1.1
 - b. 255.255.255.0
 - c. 1.0.1.0
 - d. 127.0.0.1
14. Which of the following is an alternative to configuring each workstation on a network with its own IP address?
 - a. DHCP
 - b. SNMP
 - c. RARP
 - d. TFTP
15. What kind of network operating system requires IPX/SPX?
 - a. Windows 2000 Server
 - b. UNIX
 - c. NetWare version 3.2 or lower
 - d. NetWare versions higher than 3.2
16. Which IPX/SPX core protocol provides data reliability services?
 - a. IPX
 - b. SPX
 - c. NCP
 - d. SAP

17. The node address portion of an IPX/SPX address is equivalent to what other address?
 - a. MAC address
 - b. IP address
 - c. Data Link layer address
 - d. Network address
18. What function is performed by the time to live (in IP) and the transport control (in IPX) fields?
19. Which of the following is not a valid network address for a NetWare server?
 - a. F290F45A
 - b. AAAAAAAAAA
 - c. 23AK80A3
 - d. 01010101
20. Why wouldn't you want to use NetBEUI for Internet connections (pick two reasons)?
 - a. It's not routable.
 - b. It's not secure.
 - c. It's not reliable.
 - d. It's not efficient.
21. Why are hosts on a TCP/IP network assigned host names?
22. IPX/SPX is not a routable protocol. True or False?
23. Macintosh computers can be integrated with Microsoft-based networks. True or False?
24. On a Windows 2000 workstation, how would you find your computer's NetBIOS name?
 - a. Click Start, click Run, and type ipconfig /all.
 - b. Click Start, click Settings, click Network and Dial-Up Connections, then right-click on the LAN Connection icon.
 - c. Double-click My Computer, click General, and note the computer identification text.
 - d. Click Start, click Settings, click Control Panel, then double-click the System icon. In the System Properties window, click the Identification tab.
25. Which AppleTalk protocol ensures reliable data delivery?
 - a. NCP
 - b. ZIP
 - c. DDP
 - d. ATP

26. What is a logically defined group of workstations called on an AppleTalk network?
 - a. an AppleTalk zone
 - b. an AppleTalk domain
 - c. an AppleTalk segment
 - d. an AppleTalk universe
27. On a Windows 2000 workstation, after you install the NWLink (IPX/SPX) protocol, you need not modify its network address to use it. True or False?
28. On a Windows 98 workstation, what is the default setting for the IP address in the TCP/IP protocol properties?
 - a. Obtain IP address automatically
 - b. Specify an IP address
 - c. Enable NetBIOS over TCP/IP
 - d. Enable WINS resolution
29. What information does the winipcfg command (run from a Windows 98 workstation) give you?
30. How many protocols can you install on a single Windows 98 workstation?
 - a. 2
 - b. 3
 - c. 4
 - d. as many as you want

HANDS-ON PROJECTS

You can detect protocols and test their effects through a variety of ways. The Hands-on Projects that follow add to what you have learned about protocols thus far, and form the basis for protocol troubleshooting and more in-depth analysis of the TCP/IP protocol in Chapter 11.



Project 3-1

This project requires a workstation running Windows 2000 Professional that has the TCP/IP protocol installed and that is connected to a Windows 2000 server with Internet access. It introduces the PING (Packet Internet Groper) utility, which can be used to verify that TCP/IP is running, configured correctly, and communicating with the network. A ping test is typically the first thing network professionals try when troubleshooting a TCP/IP connection problem. The process of sending out a signal is known as pinging. You can ping either an IP address or a host name. (You will learn more about PING and other diagnostic TCP/IP utilities in Chapter 11.)

1. Click **Start**, then click **Run**. The Run dialog box opens.
2. In the Open text box, type **command**, then click **OK**. The Command Prompt window opens.

3. At the DOS prompt, type **PING 127.0.0.1**. (Remember that 127.0.0.1 is the loopback address.) If your workstation is properly connected to the network, you should see a screen that contains five lines. The first line will read “Pinging 127.0.0.1 with 32 bytes of data.” Following that you will see four lines that begin “Reply from 127.0.0.1.” If you do not see four positive reply lines, or if you see four lines with the words “Request timed out,” check the syntax of your ping command. If you typed the command correctly, check the status of your TCP/IP protocol. Is it installed and bound to your NIC? To reinstall TCP/IP, follow the steps mentioned earlier in this chapter for installing protocols.
4. At the end of each of the four reply lines, a TTL value appears. What is the value of the TTL and what does this number represent? Because you received these replies to your loopback ping test, you know that your TCP/IP services are installed correctly and bound to your NIC. The loopback test, however, doesn’t indicate whether your TCP/IP services are operating correctly to grant you access to the network. In the next step, you will try a ping test that can help you determine whether your TCP/IP services are operating successfully.
5. At the DOS prompt, type **PING www.yahoo.com**.
6. What was the response? If you received a “Request timed out” message, why might you have received it? If you received a valid response, with four lines of replies, note the TTL. Why does it differ from the TTL observed when you pinged the loopback address?
7. Type **exit** at the MS DOS prompt to close the window.



Project 3-2

This project requires a Windows 2000 Professional workstation that is connected to a Windows 2000 network and has the TCP/IP protocol installed. In this project, you will uninstall the TCP/IP protocol, try the PING test again, then reinstall the TCP/IP protocol.

1. Log on to the workstation as an Administrator.
2. Click **Start**, point to **Settings**, then click **Network and Dial-up Connections**.
3. Right-click the **Local Area Connection** icon and then click **Properties** in the shortcut menu. The Properties dialog box opens, as shown in Figure 3-12.
4. Click **Internet Protocol (TCP/IP)** in the list of components, then click **Uninstall**.
5. Click **Yes** when asked to confirm the deletion.
6. You are prompted to restart your workstation to allow the changes to take effect.
7. Click **Yes** to restart your workstation.
8. When your workstation restarts, do you see any error messages? If so, write them on a separate piece of paper, then choose to ignore the errors and continue the start-up process.

9. Log on to the workstation as an Administrator again, and try pinging the loopback address as you did in Project 3-1. How did your workstation respond?

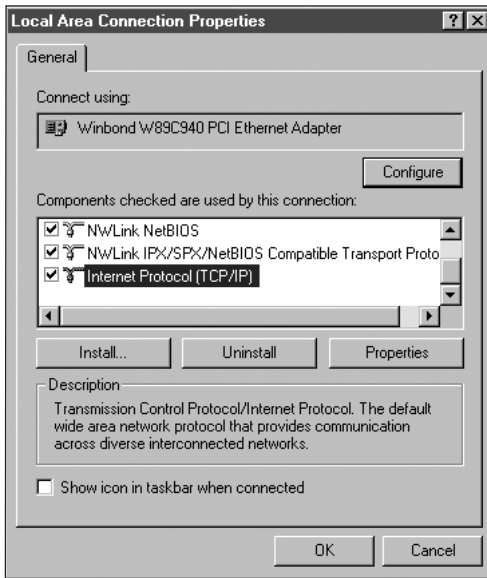


Figure 3-12 Local Area Connection Properties dialog box

10. Repeat steps 1, 2, and 3 from this project.
11. Verify that the Local Area Connection Properties dialog box is open, and then click **Install**. The Select Network Component Type dialog box opens.
12. Click **Protocol**, then click **Add**. The Select Network Protocol dialog box opens.
13. In the list of protocols, click **Internet Protocol (TCP/IP)** and click **OK**.
14. Click **Close** to install the TCP/IP protocol. The Local Area Connection Properties dialog box closes.
15. Try pinging the loopback address once more. How does your workstation respond?



Project 3-3

In this project, you will exercise your knowledge about data frames and datagrams. Because these structures form the basis of all networking, it's important that you be able to visualize and understand their components. You will need a pencil and paper to complete this project.

1. Pretend that you are a device on a network with the node address of 05 73 AC 22 and you want to send 8 bytes of data to another device on the network that has the node address 22 A0 F3 D1. You are on an Ethernet network with an external network address of 00 00 20 20. Draw a picture of the IPX/SPX datagram that will carry your data. Name the parts of the datagram and fill in the values that you can (for example, addresses, size, and packet type).

2. Pretend that you are the same device as in Step 1 sending 8 bytes of data to the same second device on the same network; this time, however, you're sending the information with the TCP/IP protocol. Your IP address is 209.122.38.7 and the second device's IP address is 209.122.38.9. Draw a picture of the TCP/IP datagram that will carry your data. Name the parts of the datagram and fill in the values that you can (for example, addresses, length, and protocol type).

CASE PROJECTS



1. As a consultant for the First National Bank of Monroe, you have been asked to solve a problem on the bank's network that began on Monday. According to the bank manager, at the beginning of each day two of the 16 tellers have been unable to log on to the network. Two other tellers occasionally experience problems at the beginning of the day, but not if they get to work before everyone else. They receive an error that says something like "another machine is using that name." When you arrive at the bank, the college intern who has been setting up the machines tells you that he is using a program called Ghost to clone all PCs from a single disk image. In other words, an exact copy of one machine's software, operating system, and its properties has been copied to all of the computers. All of the PCs are brand new, are running Windows 98, and use the same hardware and software. First National Bank's network consists of two Windows 2000 servers and runs both TCP/IP and NetBIOS/NetBEUI protocols. It uses DHCP to allocate TCP/IP addresses. What might be preventing the two tellers from logging on to the network in the morning?
2. First National Bank's president congratulates you on quickly solving the problem. She then shares the information that she's about to make an offer to buy Monroe's other bank, Metropolitan Savings. She's worried that the two banks' networks won't integrate easily. She isn't sure what kinds of servers or workstations are used by the other bank, but Metropolitan Savings' manager mentioned something about a network that relies on the Internet. What can you tell her about integrating the two systems? What protocols would you recommend that she use or continue to use to facilitate the integration process?
3. Six months later, First National Bank has successfully consolidated the networks at its original location and at its new acquisition. Business is booming, and the bank is investigating the possibility of allowing customers to check their account balances from the Web. However, the bank's president tells you the bank doesn't have its own connection to the Internet at this time. She understands that she needs to obtain IP addresses for all of her machines. But, she says, they are already using IP addresses internally and they work well without having to pay ICANN for new IP addresses. Would you recommend leaving the bank's IP addressing as is or changing it? How do you suggest that the bank obtain Internet access? What concerns would you bring up with regard to allowing customers access to their account information off the Web? How might Internet access affect the bank's internal LAN?